

APPLICATION FOR UNITED STATES LETTERS PATENT

INVENTORS: Sang Wook KIM and Young Wook JUN

TITLE: A METHOD FOR PROCESSING DYNAMIC DATABASE IN
DISTRIBUTED PROCESSING SYSTEM BASED ON CORBA
PLATFORM

ATTORNEYS: FLESHNER & KIM, LLP
&
ADDRESS: P. O. Box 221200
Chantilly, VA 20153-1200

DOCKET NO.: P-0316

A METHOD FOR PROCESSING DYNAMIC DATABASE IN DISTRIBUTED PROCESSING SYSTEM BASED ON CORBA PLATFORM

BACKGROUND OF THE INVENTION

1. Field of the Invention

[1] The present invention relates to a method for processing a database in a distributed processing system and, more particularly, to a method for processing a dynamic data base in a distributed processing system, based on a Common Object Request Broker Architecture (CORBA) platform.

2. Background of the Related Art

[2] An Object Request Broker (ORB) is a virtual system for communicating between objects in a distributed object environment, and CORBA is a standard specification of the ORB. ORB is arranged by an Object Management Group (OMG) to standardize the object oriented technique.

[3] Generally, a client that requests a service and an object implementation, providing the service, give and take a request or a service result (returned value) through the ORB, in the distributed object environment, and then the process is progressed. To begin with, the client passes a name and request content of the object, which is to be served, to the ORB. Then, the ORB finds the object implementation, based on the name, and transmits the requested contents. The object implementation returns a result of the

service performance (return value) to the ORB, and the ORB transmits the result to the client. The client should be compatible with the interface of the ORB, to use a service of the optional object implementation. In addition, an Interface Definition Language (IDL) is used for defining the interface.

[4] A platform is characterized as the hardware or software that forms the basis of a computer system. That is, an Operation System (OS) or an application is operated in a hardware or software environment.

[5] Figure 1 illustrates database processing in a background art distributed processing system, based on the CORBA platform. A process is shown in which a client 10 accesses a database 30, of a server 20, and obtains data through a distributed system, based on the CORBA platform. Although it is not shown in Figure 1, the server 20 consists of a plurality of objects, and there is an object that processes the database in the plurality of objects. The object processing the database includes a plurality of methods.

[6] As shown in Figure 1, when the client 10 calls the method on the server 20, the corresponding method detects the data, wanted by the client 10, in the database 30. The server 20 then returns the data to the client 10 through an Internet Inter-ORB Protocol (IIOP) 40. According to the method, various methods are required because there are various requests by the client 10 for the data.

[7] There are two parameters that are required when the client 10 calls the method. These are a Structure Query Language (SQL) string, for detecting data of the

database 30, and a pointer of a data structure, for allocating the detected data. A transmission and a return of the parameters are made by a “call by reference” method. The SQL string is a query for the data that is wanted, and the data structure defines a structure and a formation of a storing space for storing the detected data.

[8] The call by reference method is a method not for giving and taking a real value or data, but for giving and taking a pointer of a memory area, in which a real value or data is stored. And the above-mentioned pointer may be referred to as a reference pointer.

[9] Figure 2 is an exemplary view showing an IDL for the call by reference method. Generally, the client 10 finds a reference pointer of the database process object in the CORBA platform server 20. Then, the client 10 calls a method of the server 20 through the reference pointer. The called method detects the wanted data in the database 30 and then constructs the detected data into the data structure form, defined by the IDL, to transmit it to the client 10. In addition, the data structure is defined in the respective method.

[10] If the method called by the client 10 does not exist or does not satisfy the client’s request, the server 20 defines a new method. Thereafter, the server 20 should compile the IDL that is coded after adding the newly defined method to the previously defined methods.

[11] Figure 3 is a flow chart showing a database processing method using the background art CORBA platform. As shown therein, the client 10 finds the reference pointer of the database processing object in the server 20 (S1) and calls the method of the server 20 through the reference pointer (S2).

[12] The called method detects the data required by the client 10, in database 30, through a search condition of the SQL string. Also, the called method allocates the storing spaces, according to the data structure form defined by the client 10 (S3 and S4), and copies the detected data to the allocated storing space (S5). In addition, a result value calculated by the above processes (S3, S4, and S5) is transmitted to the client 10, through the ORB 40 (S6).

[13] An embodiment for processing the database using the background art distributed processing system, based on the CORBA platform, will be described in more detail with reference to Figures 2 and 3. The client 10 finds a reference pointer of the database processing object, located in the server 20, to detect the wanted data, by accessing the database 30. After that, the client 10 calls the corresponding method using the SQL string and the data structure pointer as method calling parameters. The SQL string is a parameter related to the query of the client. The called method detects the corresponding data using the SQL string as a database searching condition. The server 20 allocates the storing space according to the data structure form defined by the called method and copies the detected data to the allocated storing space.

[14] The result value calculated through the above processes is stored in a certain memory area of the server 20 and returned to the memory area of the client 10, pointed to by the data structure pointer, through ORB 40. For example, assume that items of “name”, “customer number”, “birthday”, and “customer score” of customers having a customer score greater than 500 points are detected within the database by the client 10, among various information related to the customers stored and managed in database 30. Further assume that the detected data are “AAA”, “1011”, “05-27-1977”, “501”, “BBB”, “1120”, “11-20-1972”, “620”, “CCC”, “1313”, “10-28-1974”, and “550”.

[15] The method-A IDL call of client 10 is shown in Table 1, below.

Table 1.
<u>Step A</u> Struct structA { string strA; /"name" item/ short intA1; /"customer" item/ date datA2; /"birthday" item/ short intA3; /"customer score" item/ };
<u>Step B</u> typedef sequence < structA > returndataA;
<u>Step C</u> short methodA(in string SQLstring, out returndataA retVal);

[16] Step A defines the data structure that will be detected in database 30. Step B declares the data structure, defined in step A, as a sequence type. Step C is a sentence for calling a method having two parameters. As described above, if the data structure is declared as the sequence type, arrays having the same number of stored data are generated, so that memory is not wasted.

[17] The client 10 creates the data structure reference pointer and the SQL string parameters and calls the corresponding method in the server 20 (Step C). The called method detects the corresponding data in the database 30 and allocates the storing space, in the form defined by the data structure. In addition, the method copies the detected data to the allocated storing space.

[18] The result value, calculated through the above processes, is stored in a certain memory area of the server 20. After that, the result is returned to the memory area of the client 10, pointed to by the data structure pointer, through the ORB 40. In addition, the client 10 accesses the memory area in which the returned value is stored and outputs the corresponding data.

[19] As described above, the server 20 should add a new method to the database processing objects, when no current method is able to accept the request of the client 10. In addition, when the new method is added, both the original methods and the new method should be compiled to redefine the IDL. As new methods are added, the corresponding objects get larger.

[20] Also, in the process of designing the database program, the developer should consider all prospective client requests and should define a plurality of methods and data structures to support these client requests. Therefore, it takes a lot of time and expense to develop the program.

[21] The above references are incorporated by reference herein where appropriate for appropriate teachings of additional or alternative details, features and/or technical background.

SUMMARY OF THE INVENTION

[22] An object of the invention is to solve at least the above problems and/or disadvantages and to provide at least the advantages described hereinafter.

[23] Therefore, an object of the present invention is to provide a method for processing a dynamic database in a distributed processing system, based on CORBA platform, that defines data structures, processed between a client and a server, and allocates memory for the structures automatically.

[24] Another object of the present invention is to provide a method for processing a dynamic database in a distributed processing system, based on CORBA platform, by which requests of the client can be accepted using a method.

[25] To achieve these and other advantages and in accordance with the purpose of the present invention, as embodied and broadly described herein, there is provided a

method for processing a dynamic database in a distributed processing system, based on a CORBA platform, including calling a generic method of a server; detecting data using the called generic method; storing the detected data in a local memory; generating or expanding a storing space, according to the form and number of the detected data; copying the detected data in the local memory, in a storing space generated in a certain memory area of the server; and returning the detected data to a client.

[26] Additional advantages, objects, and features of the invention will be set forth in part in the description which follows and in part will become apparent to those having ordinary skill in the art upon examination of the following or may be learned from practice of the invention. The objects and advantages of the invention may be realized and attained as particularly pointed out in the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[27] The invention will be described in detail with reference to the following drawings in which like reference numerals refer to like elements wherein:

[28] Figure 1 illustrates database processing in the background art distributed processing system, based on a CORBA platform;

[29] Figure 2 illustrates an Interface Definition Language (IDL) for several methods;

[30] Figure 3 illustrates a flow chart showing the background art method for database processing in the distributed processing system, based on the CORBA platform;

[31] Figure 4 illustrates a method for processing a dynamic database according to the present invention;

[32] Figure 5 illustrates an IDL for a preferred method of processing a dynamic database, according to the present invention; and

[33] Figure 6 illustrates a flow chart showing the method for processing the dynamic database in the distributed processing system, based on the CORBA platform, according to the present invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

[34] Figure 4 is a block diagram showing a method for processing a dynamic database in a distributed processing system, based on a Common Object Request Broker Architecture (COBRA) platform, according to the present invention. A form of data detected in the database is declared as an Any type, and a structure of the data is declared as a sequence type. Various requests of clients can be accepted using a method. The Any type may take any one of numerous data forms, such as string, integer, float, decimal, and date, and is able to present all of the data forms used in the database 30. In addition, a memory space declared as the sequence type has basically the same structure as that of an array structure. However, it is different in that the memory space is generated

automatically to have the same number of bits (i.e., size) as that of the data. If the storing space, which has been declared once, is declared as the sequence type again, the storing space is expanded as two-dimensional space and then expanded to n-dimensional space, if the space is declared as the sequence type n times.

[35] The method for processing a dynamic database in a distributed processing system, based on a CORBA platform, according to the present invention comprises calling a generic method of a server 20; detecting data by the called generic method; storing the detected data in a local memory; generating or expanding a storing space according to the form and number of the detected data; copying the detected data in a local dynamic memory and in a storing space generated in a certain memory area of the server 20; and returning the detected data to a client 10.

[36] Figure 6 is a flow chart showing the method for processing the dynamic database in the distributed processing system, based on the CORBA platform, according to the present invention. The operation of the present invention will be described, as follows, with reference to Figure 6. The client 10 finds a reference pointer of a database processing object (S11) and calls the generic method in the server 20, using the SQL string and a pointer of the generic data structure as parameters (S12). In defining the data structure, the client 10 does not define the data type or structure, but defines a generic data structure using declarative functions of Any type and of sequence type, so that the

memory area is automatically generated according to the structure or the number of the detected data.

[37] The called generic method detects the corresponding data in the database 30, under the searching condition of the SQL string (S13). The detected data are stored in the local dynamic memory (S14). In addition, the generic method generates a storing space by instantiating the data structure. After that, the data stored in the local memory are copied to the generated storing space (S16 and S17).

[38] The processes S13 ~ S17 are repeatedly performed until all the data satisfying the SQL string (Query) are detected (S18). When the processes S13 ~ S17 are completed, a result value in the storing space is returned to the client 10, in the same manner as in the background art (S19).

[39] In process S15, the detected data are stored in the local memory in a linked-list form, regardless of the data table structure. To return the detected data to the client 10, the generic method allocates the storing spaces as the number of the data using the sequence type declarative function and declares the respective storing spaces as Any types. This is referred to as an instantiation process.

[40] Processing procedures for process S11 and process S19 are the same as the processing method in the background art distributed processing system, based on the CORBA platform.

[41] Figure 5 is a preferred embodiment of an Interface Definition Language (IDL) method according to the present invention. The method for processing a dynamic database in the distributed system, based on the CORBA platform, of the present invention will be described with reference to Figure 5.

[42] Assume the client 10 wants to detect items of "name", "customer number", "birthday", and "customer score" of the customers having customer scores greater than 500 points, among the various information related to the customers stored and managed in database 30. Further assume that the detected data are "AAA", "1101", "05-27-1977", "501", "BBB", "1120", "11-20 1972 ", "620", "CCC", "1313", "10-28-1974", and "550".

[43] A generic method calling IDL of the client 10 is provided in Table 2, below.

Table 2.
<u>Step AA</u> Struct elementValue { short type; /1:integer 2:float 3:string 4:decimal 5:date/ any value; /any type for real value/ };
<u>Step BB</u> typedef sequence <elementValue> ElemSeq;
<u>Step CC</u> typedef sequence <ElemSeq> ReturnSeq;
<u>Step DD</u> short genericMethod(in string SQLstring, out ReturnSeq RetVal);

[44] Step AA defines the generic data structure so that the detected data has the Any type data form. Step BB declares the data structure defined in step AA as the sequence type and generates an array of first structure. In addition, step CC declares the data structure defined in step BB as the sequence type again, and generates the array of a two dimensional structure. Step DD shows the calling for the generic method.

[45] To begin with, the client 10 calls the generic method by setting the “pointer of the generic data structure” and the “SQL string” as parameters, which are respectively defined as Any type and sequence type. The called generic method applies the SQL string

through a dynamic SQL mode, supported by database 30. Then, the corresponding data is detected and stored in the local memory in a linked-list form.

[46] The first row data is all detected and the generic method instantiates the data structure in a certain memory area of the server 20, whereby a storing space for copying the row data is generated (Step BB). The storing space generated by the instantiation is comprised of four arrays declared as Any types, and the four elements of detected data ("AAA", "1011", "05-27-1977", and "501") comprising the first row will be stored in the four respective arrays, as shown by Table 3.

Table 3.				
	<u>Array 1</u>	<u>Array 2</u>	<u>Array 3</u>	<u>Array 4</u>
Type	Any type	Any type	Any type	Any type
Declaration				
Stored Value	AAA	1011	05-27-1977	501

[47] When the generic method detects all of the data satisfying the SQL string (Query), that is, when the second row data ("BBB", "1120", "11-20 1972 ", "620") and the third row data ("CCC", "1313", "10-28-1974", and "550") are detected, the one-dimensional array is declared once again as the sequence type (Step CC) and the two-dimensional storing space is generated as shown by Table 4.

Table 4.				
	<u>Array 1</u>	<u>Array 2</u>	<u>Array 3</u>	<u>Array 4</u>
First Row Data	Any Type	Any Type	Any Type	Any Type
First Row Data	Any Type	Any Type	Any Type	Any Type
First Row Data	Any Type	Any Type	Any Type	Any Type

[48] Of course, the detected respective data are stored in the local memory and then copied to the memory area as a row-data unit. In addition, whenever the row data are copied to the memory area, four storing areas (arrays), which are declared as Any type, are expanded to store the data.

[49] The storing areas (3*4 arrays) generated in the memory area store the data, as shown by Table 5.

Table 5.				
First Row Data	AAA	1011	05-27-1977	501
Second Row Data	BBB	1120	11-20-1972	620
Third Row Data	CCC	1313	10-28-1974	550

[50] As described above, the data structure defined by the generic method is generated and expanded dynamically by the Any type and the sequence type. That is, the data structure of the generic method is not defined to have a fixed form, before calling the method, but is flexibly generated according to the form and the number of the detected data, after searching the database 30.

[51] When all of the corresponding data are detected, the detected result value is stored in the memory area of the client 10, pointed to by the pointer of the data structure, through the ORB. The client 10 accesses the memory area, in which the result value is stored, and outputs the corresponding data.

[52] As described above, according to the present invention, the method IDL is defined generically, and the defined method automatically generates and expands the memory area in which the detected data is stored. Thereby, a range of program designing is minimized. Therefore, the developer of the program is able to reduce the time and expense spent defining the IDL of the plurality of methods and defining the data structure.

[53] Also, for a transaction request of the client that is not defined, the background art defines new IDLs and must then compile them. However, according to the present invention, various transaction requests of the client can be accepted using the generic method and the generic data structure, which are defined when the database program is developed.

[54] Also, according to the present invention, the problem of dynamic binding of the data (i.e., a form or a structure of the data cannot be known until the data is detected) generated while processing the SELECT query, of the background art database, can be solved. Thereby, processing for various distributed transactions, based on the CORBA platform, and the function of the processor can be increased, accordingly.

[55] The foregoing embodiments and advantages are merely exemplary and are not to be construed as limiting the present invention. The present teaching can be readily applied to other types of apparatuses. The description of the present invention is intended to be illustrative, and not to limit the scope of the claims. Many alternatives, modifications, and variations will be apparent to those skilled in the art. In the claims, means-plus-function clauses are intended to cover the structures described herein as performing the recited function and not only structural equivalents but also equivalent structures.